

Corso di Ingegneria del Software Deliverable di progetto	2024-2025
---	-----------

“Ingegneria del Software” 2024-2025

Docente: Prof. Angelo Furfaro

<Sistema di gestione di un Organigramma>

Data	<14/01/2026>
Documento	Documento Finale – D3

Team Members		
Nome e Cognome	Matricola	E-mail address
Antonio Laudonio	209527	ldnntn01d10b774i@stud enti.unical.it

Sommario

- **Descrizione del contesto applicativo e dello stato dell'arte relativo ai sistemi di gestione degli organigrammi aziendali**
- **Analisi e affinamento dei requisiti funzionali e non funzionali del sistema**
- **Descrizione dettagliata degli scenari d'uso principali**
- **Individuazione dei requisiti esclusi e delle assunzioni di progetto**
- **Presentazione dell'architettura software e delle principali scelte progettuali**
- **Modellazione dei dati e descrizione della persistenza delle informazioni**
- **Progettazione di basso livello delle classi di dominio**
- **Spiegazione di come il sistema soddisfa i requisiti funzionali e non funzionali**
- **Descrizione e validazione del prototipo realizzato, con riferimento alle funzionalità implementate**

List of Challenging/Risky Requirements or Tasks

<i>Challenging Task</i>	<i>Date the task is identified</i>	<i>Date the challenge is resolved</i>	<i>Explanation on how the challenge has been managed</i>
Analisi e comprensione della traccia di progetto	20/12/2025	22/12/2025	La traccia è stata analizzata in modo approfondito per identificare i requisiti funzionali e non funzionali. Sono stati chiariti obiettivi, vincoli e ambito del sistema.
Studio dell'architettura backend e dei principi di progettazione	27/12/2025	05/01/2026	È stato svolto uno studio delle architetture backend e dei servizi REST, al fine di strutturare correttamente il sistema e separare le responsabilità applicative.
Modellazione dei dati e delle relazioni gerarchiche	05/01/2026	10/01/2026	La complessità della struttura gerarchica è stata gestita tramite una modellazione ad albero, definendo relazioni chiare tra le unità organizzative.
Persistenza dei dati e gestione dello stato applicativo	10/01/2026	13/01/2026	È stata introdotta una gestione persistente dei dati tramite DBMS, assicurando la consistenza delle informazioni anche dopo il riavvio del sistema.
Progettazione e realizzazione dell'interfaccia grafica (GUI)	08/01/2026	14/01/2026	L'interfaccia utente è stata progettata prendendo spunto da soluzioni affrontate in altri corsi, con particolare attenzione a usabilità, chiarezza e coerenza visiva.

A. Stato dell'Arte

La gestione dell'organigramma aziendale è una funzionalità comune a molti sistemi di gestione delle risorse umane e piattaforme ERP, che consentono di rappresentare strutture organizzative, ruoli e relazioni gerarchiche. Tali soluzioni risultano spesso complete ma complesse e poco flessibili, specialmente per contesti di piccole e medie dimensioni.

Le moderne applicazioni web adottano architetture client-server basate su servizi REST, che permettono una gestione modulare e una rappresentazione efficace di strutture gerarchiche tramite modelli ad albero persistenti. In questo scenario, emerge la necessità di strumenti più semplici e mirati.

Il progetto si inserisce in questo contesto proponendo un sistema dedicato esclusivamente alla gestione dell'organigramma aziendale, con particolare attenzione a semplicità, chiarezza e usabilità, prendendo spunto dalle soluzioni esistenti ma adattandole a un contesto didattico e progettuale.

B. Raffinamento dei Requisiti

A.1 Servizi (con prioritizzazione)

S1 - Gestione delle Unità Organizzative

Importanza: Alta

Complessità: Media

Il sistema deve consentire la creazione, modifica ed eliminazione di unità organizzative.

Le unità possono essere strutturate in forma gerarchica, permettendo l'associazione di unità figlie a un'unità padre. Deve essere garantita la coerenza delle relazioni gerarchiche e l'integrità della struttura dell'organigramma.

S2 – Gestione dei Dipendenti**Importanza: Media****Complessità: Bassa**

Il sistema deve permettere la creazione e la gestione dei dipendenti. Un dipendente può essere associato a più unità organizzative e può ricoprire ruoli differenti a seconda dell'unità di appartenenza. L'assegnazione di un ruolo è possibile solo se tale ruolo è definito tra quelli ammessi dall'unità organizzativa.

S3 – Gestione dei Ruoli**Importanza: Media****Complessità: Media**

Il sistema deve consentire la definizione e la gestione dei ruoli associabili alle unità organizzative. Ogni unità può avere un insieme di ruoli ammessi, che rappresentano le posizioni ricopribili dai dipendenti all'interno di quella specifica unità.

S4 – Visualizzazione Grafica dell'Organigramma**Importanza: Alta****Complessità: Alta**

Il sistema deve fornire una rappresentazione grafica dell'organigramma aziendale. Ogni unità organizzativa deve essere visualizzata come un oggetto grafico posizionato su un pannello, mentre le relazioni gerarchiche tra unità padre e figlio devono essere rappresentate tramite collegamenti grafici. La visualizzazione deve aggiornarsi dinamicamente in seguito a modifiche della struttura organizzativa.

S5 – Interazione e Navigazione**Importanza: Media****Complessità: Media**

Il sistema deve consentire all'utente di interagire con l'organigramma, selezionando unità organizzative per visualizzarne i dettagli, i dipendenti assegnati e i ruoli disponibili. L'interfaccia deve supportare una navigazione intuitiva e facilitare la comprensione della struttura gerarchica.

A.2 Requisiti non Funzionali

NFR1 – Correttezza

Il sistema deve garantire la correttezza delle operazioni effettuate dall'utente, impedendo l'inserimento di dati non validi. In caso di operazioni non consentite o parametri errati, il sistema deve fornire messaggi di avviso chiari e comprensibili.

NFR2 – Usabilità

Il sistema deve offrire un'interfaccia grafica intuitiva e di facile utilizzo. Le azioni non disponibili in un determinato contesto devono essere disabilitate automaticamente, in modo da guidare l'utente ed evitare errori durante l'interazione con l'organigramma.

NFR3 – Affidabilità

Il sistema deve garantire un comportamento stabile durante l'utilizzo, mantenendo la coerenza della struttura dell'organigramma anche in presenza di modifiche frequenti, come l'aggiunta o la rimozione di unità organizzative e dipendenti.

NFR4 – Manutenibilità

Il sistema deve essere progettato in modo modulare, facilitando eventuali estensioni future, come l'aggiunta di nuovi servizi o funzionalità, senza richiedere modifiche significative all'architettura esistente.

A.3 Scenari d'uso dettagliati

Scenario 1 – Creazione Unità Radice

Attore principale: Utente

Pre-condizione: Il sistema è avviato.

Flusso principale

1. L'utente clicca su "+ Radice".
2. Il sistema mostra un modulo di inserimento.
3. L'utente inserisce il nome dell'unità (obbligatorio).
4. L'utente conferma.
5. Il sistema crea l'unità radice e la visualizza nell'area "Organigramma".

Estensioni

3a. Se il nome è vuoto o non valido, il sistema mostra un avviso e blocca la conferma.

Post-condizione: Esiste una nuova unità radice nell'organigramma.

Scenario 2 – Selezione di un'unità e visualizzazione dettagli

Attore principale: Utente

Pre-condizione: Esiste almeno un'unità nell'organigramma.

Flusso principale

1. L'utente clicca su un nodo dell'organigramma.

2. Il sistema evidenzia il nodo selezionato.
3. Il pannello “Dettaglio” mostra informazioni dell’unità (nome, ID, numero figli) e i comandi disponibili.

Post-condizione: L’unità selezionata è pronta per operazioni (aggiunta sotto-unità, ruoli, assegnazioni, eliminazione).

Scenario 3 – Aggiunta di una sotto-unità

Attore principale: *Utente*

Pre-condizione: *Un’unità è selezionata.*

Flusso principale

1. L’utente clicca su “+ Aggiungi sotto-unità”.
2. Il sistema mostra un modulo di inserimento.
3. L’utente inserisce il nome della nuova unità (obbligatorio).
4. L’utente conferma.
5. Il sistema crea la sotto-unità e la collega come figlia dell’unità selezionata.
6. L’organigramma viene aggiornato mostrando il nuovo nodo.

Estensioni

- 3a. Se il nome è vuoto/non valido, il sistema mostra un avviso e non crea l’unità.

Post-condizione: La struttura padre–figlio è aggiornata e consistente.

Scenario 4 – Gestione dei ruoli ammessi di un’unità

Attore principale: *Utente*

Pre-condizione: *Un’unità è selezionata.*

Flusso principale

1. L’utente clicca su “Gestisci ruoli ammessi”.
2. Il sistema mostra una lista di ruoli disponibili con checkbox e l’opzione di creazione di nuovi ruoli.
3. L’utente:
 - crea un nuovo ruolo oppure
 - seleziona/deseleziona ruoli già esistenti tramite checkbox.
4. L’utente conferma le modifiche.
5. Il sistema aggiorna l’insieme dei ruoli ammessi per l’unità.

Estensioni

3a. Se si tenta di creare un ruolo con nome vuoto/duplicato, il sistema mostra un avviso e non lo aggiunge.

Post-condizione: L'unità ha un insieme aggiornato di ruoli ammessi.

Scenario 5 – Creazione di un nuovo dipendente

Attore principale: *Utente*

Pre-condizione: *Il sistema è avviato.*

Flusso principale

1. L'utente (nel pannello sinistro "Registro dipendenti") clicca su "+ Nuovo".
2. Il sistema mostra un modulo di inserimento.
3. L'utente inserisce nome e cognome (obbligatori).
4. L'utente conferma.
5. Il sistema crea il dipendente e lo mostra nella lista.

Estensioni

3a. Se nome/cognome mancanti o non validi, il sistema segnala l'errore.

Post-condizione: Il dipendente è disponibile per assegnazioni nelle unità.

Scenario 6 – Assegnazione di un dipendente a un'unità con ruolo

Attore principale: *Utente*

Pre-condizione:

-Un'unità è selezionata

-Esiste almeno un dipendente nel registro

-L'unità ha almeno un ruolo ammesso (o selezionabile)

Flusso principale

1. L'utente clicca su "Assegna dipendente".
2. Il sistema mostra un form di assegnazione.
3. L'utente seleziona un dipendente dalla lista.
4. L'utente seleziona un ruolo tra quelli ammessi dall'unità.
5. L'utente conferma.
6. Il sistema registra l'assegnazione e la mostra nella sezione "Assegnazioni in questa unità".

Estensioni

4a. Se il ruolo non è ammesso per l'unità, l'assegnazione non è consentita.

Post-condizione: Il dipendente risulta assegnato all'unità con il ruolo scelto.

Scenario 7 – Eliminazione di un'unità

Attore principale: Utente

Pre-condizione: Un'unità è selezionata.

Flusso principale

1. L'utente clicca su "Elimina unità".
2. Il sistema verifica i vincoli (assenza di figli e di dipendenti associati).
3. Se i vincoli sono soddisfatti, il sistema elimina l'unità e aggiorna l'organigramma.

Estensioni

2a. Se l'unità ha figli o assegnazioni, il sistema impedisce l'eliminazione e mostra un avviso.

Post-condizione: L'unità è rimossa e la struttura resta consistente.

Scenario 8 – Eliminazione di un dipendente con rimozione automatica delle assegnazioni

Attore principale: Utente

Pre-condizione: Esiste almeno un dipendente.

Flusso principale

1. L'utente seleziona un dipendente dalla lista a sinistra.
2. L'utente clicca su elimina.
3. Il sistema elimina il dipendente dal registro.
4. Il sistema rimuove automaticamente tutte le sue assegnazioni dalle varie unità.
5. Le viste (lista dipendenti e dettagli unità) vengono aggiornate.

Post-condizione: Il dipendente non esiste più e nessuna unità mantiene riferimenti/assegnazioni verso di lui.

A.4 Excluded Requirements

Alcuni requisiti sono stati esclusi dall'implementazione finale del sistema principalmente a causa di vincoli di tempo e per mantenere la semplicità dell'interfaccia.

- Modifica di un'unità organizzativa esistente

Il sistema non consente la modifica delle informazioni di un'unità organizzativa già creata. L'aggiunta di questa funzionalità avrebbe richiesto l'introduzione di una finestra dedicata alla modifica oppure meccanismi di interazione più complessi, non implementati per limitazioni temporali.

- Modifica dei dati di un dipendente

Non è prevista la possibilità di modificare i dati anagrafici di un dipendente già creato. Eventuali cambiamenti richiedono la rimozione e la successiva ricreazione del dipendente.

- Creazione esplicita del padre di un'unità

Il sistema non consente la selezione manuale dell'unità padre durante la creazione di una nuova unità. L'unità padre viene determinata implicitamente in base all'unità selezionata dall'utente al momento dell'operazione, semplificando l'interazione e riducendo la possibilità di errori.

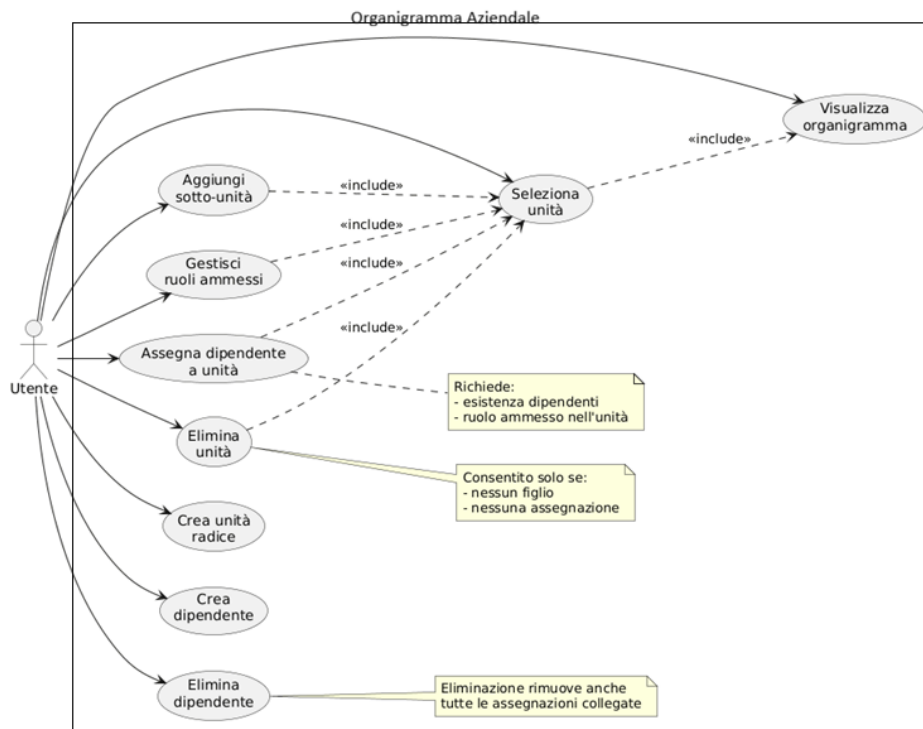
A.5 Assunzioni

Durante la progettazione e lo sviluppo del sistema sono state fatte le seguenti assunzioni:

- Il sistema viene utilizzato da un utente singolo o da utenti con competenze di base nell'utilizzo di applicazioni informatiche, pertanto non è stata implementata una gestione avanzata degli utenti o dei permessi di accesso.

- Ogni unità organizzativa è identificata in modo univoco all'interno dell'organigramma e può avere un solo padre, ad eccezione dell'unità radice.
- La struttura dell'organigramma è considerata aciclica, evitando la creazione di cicli nelle relazioni gerarchiche.
- Un dipendente può essere associato a più unità organizzative, ricoprendo ruoli differenti in ciascuna di esse, purché tali ruoli siano ammessi dall'unità.
- Si assume che i dipendenti e le unità organizzative vengano creati e gestiti correttamente dall'utente, seguendo i vincoli imposti dall'interfaccia grafica.
- Il sistema è progettato per contesti aziendali di piccole e medie dimensioni, e non è ottimizzato per gestire un numero molto elevato di unità o dipendenti.

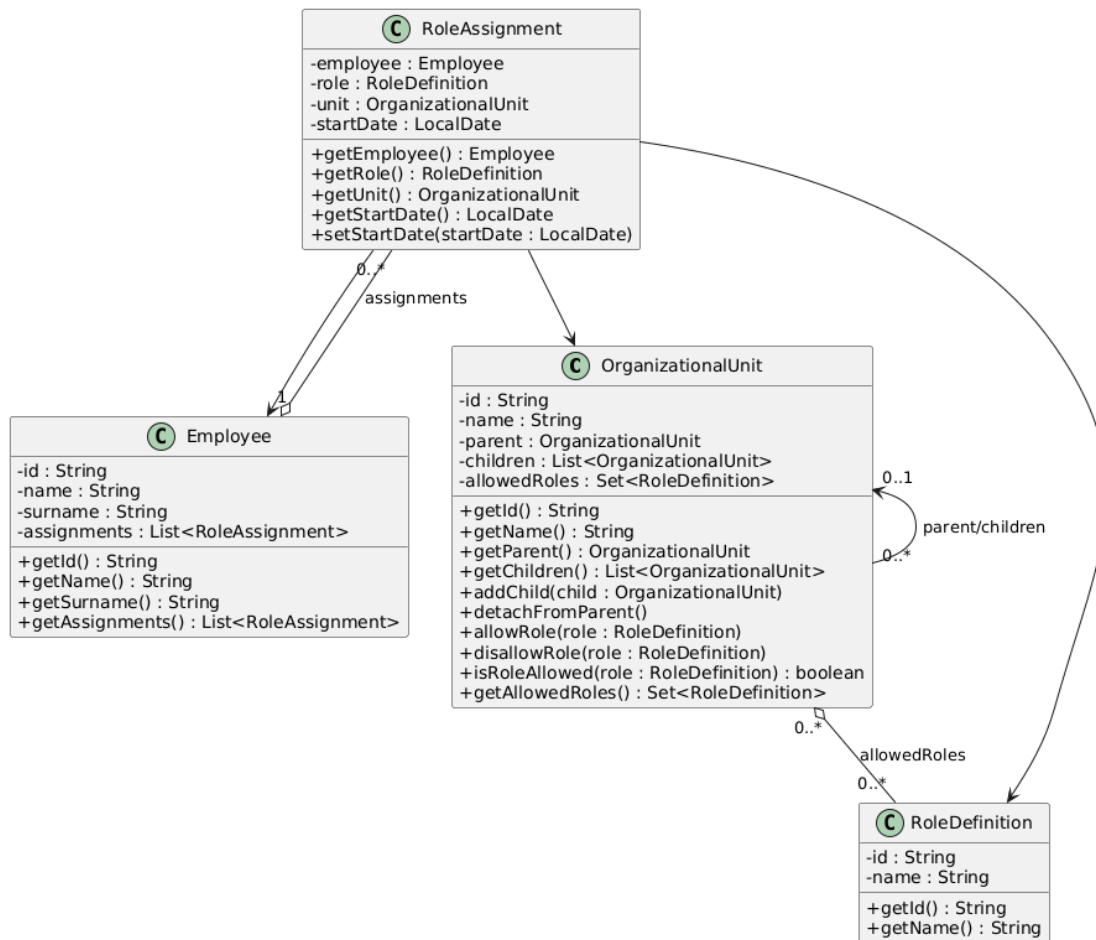
A.6 Use Case Diagrams



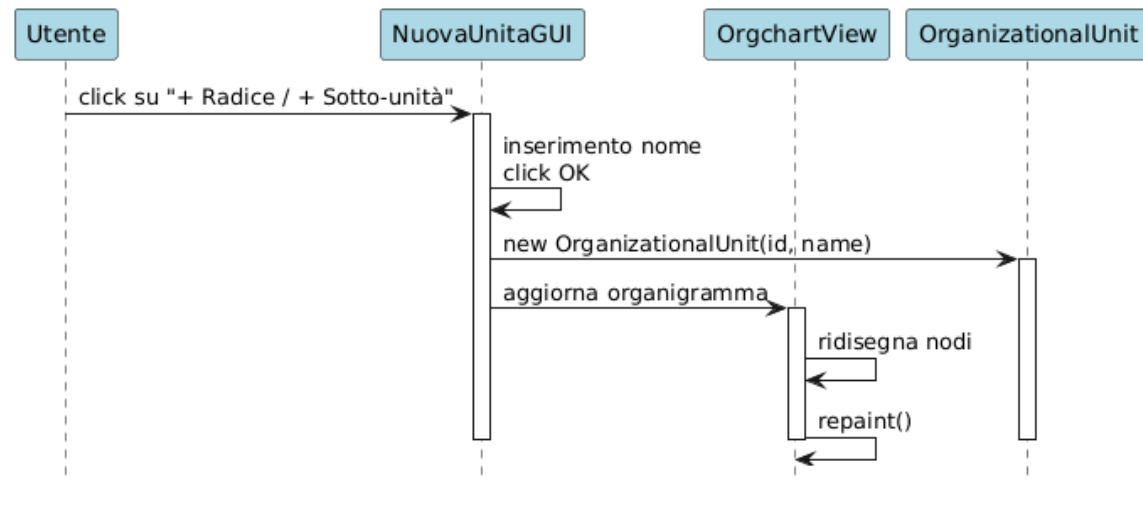
C. Architettura Software

<IF RELEVANT, Report here both the static and the dynamic view of your system design, in terms of a Component Diagram, and their related Sequence Diagrams >

C.1 The static view of the system: Component Diagram



*C.2 The dynamic view of the software architecture: Sequence
Diagram*



D. Dati e loro modellazione (se il sistema si interfaccia con un DBMS)

La realizzazione dei servizi richiesti dalla traccia necessita della memorizzazione persistente delle informazioni relative all'organigramma aziendale.

In particolare, il sistema salva:

- le unità organizzative, con il loro identificativo, nome e relazione gerarchica padre-figlio;
- i ruoli definiti nel sistema e i ruoli ammessi per ciascuna unità;
- i dipendenti, identificati da nome e cognome;
- le assegnazioni, che associano un dipendente a una unità con uno specifico ruolo.

Tali informazioni sono sufficienti per ricostruire l'intera struttura dell'organigramma e le relazioni tra unità, dipendenti e ruoli in seguito al riavvio dell'applicazione.

I dati sono memorizzati tramite un DBMS persistente (H2 su file), mentre la modellazione segue uno schema di tipo E-R, con relazioni uno-a-molti per la struttura gerarchica delle unità e relazioni molti-a-molti per l'associazione tra unità e ruoli e tra dipendenti, unità e ruoli.

I dati non sono forniti da applicativi esterni, ma vengono interamente gestiti dal sistema.

E. Scelte Progettuali (Design Decisions)

<Document here the 5 most important design decisions you had to take. You can use both a textual or a diagrammatic specification.>

1. Modellazione della struttura dell'organigramma

Per rappresentare la struttura gerarchica dell'organigramma aziendale è stata adottata una struttura ad albero, in cui ogni unità organizzativa può avere una sola unità padre e più unità figlie. Questa scelta consente di modellare in modo naturale le relazioni organizzative e di garantire l'assenza di cicli nella gerarchia.

2. Separazione tra logica di dominio e interfaccia utente

È stata scelta un'architettura client-server, separando nettamente la logica di dominio e di business dal frontend. Il backend si occupa della gestione delle unità, dei dipendenti, dei ruoli e delle assegnazioni, mentre il frontend si limita alla visualizzazione e all'interazione con l'utente. Questa separazione migliora la manutenibilità e la chiarezza del sistema.

3. Gestione delle regole di assegnazione dei ruoli

La verifica dei ruoli ammessi per una unità è stata centralizzata in un componente dedicato alla gestione delle politiche di assegnazione. In questo modo, il controllo che un ruolo sia valido per una determinata unità viene effettuato in un unico punto, evitando duplicazioni di logica e riducendo la possibilità di errori.

4. Persistenza dei dati tramite DBMS

Per garantire il mantenimento delle informazioni anche dopo il riavvio dell'applicazione, è stata scelta una soluzione di persistenza su database. Tutti i dati relativi all'organigramma, ai dipendenti, ai ruoli e alle assegnazioni vengono salvati in modo persistente, permettendo il ripristino automatico dello stato del sistema.

5. Gestione delle eliminazioni e dei vincoli di consistenza

Sono stati introdotti vincoli progettuali per preservare la consistenza dei dati. In particolare, un'unità organizzativa non può essere eliminata se possiede sotto-unità o assegnazioni attive, mentre l'eliminazione di un dipendente comporta automaticamente la rimozione di tutte le sue assegnazioni. Questa scelta evita stati inconsistenti e semplifica la gestione dei dati.

F. Progettazione di Basso Livello

La progettazione di basso livello definisce le principali classi del dominio e le loro responsabilità. In particolare, sono state progettate classi per rappresentare le unità organizzative, i dipendenti, i ruoli e le assegnazioni, con metodi dedicati alla gestione delle relazioni gerarchiche e dei vincoli di consistenza.

Le interazioni tra le classi sono state progettate in modo da garantire la correttezza delle operazioni fondamentali (creazione, eliminazione e assegnazione), mantenendo una struttura semplice e coerente con i requisiti funzionali del sistema.

G. Spiegare come il progetto soddisfa i requisiti funzionali (FRs) e quelli non funzionali (NFRs)

I requisiti funzionali del sistema sono soddisfatti attraverso una chiara separazione tra interfaccia utente, logica applicativa e modello di dominio. Le funzionalità di creazione delle unità organizzative, gestione dei dipendenti, definizione dei ruoli e assegnazione dei dipendenti alle unità sono implementate tramite servizi applicativi dedicati che operano sui modelli di dominio, garantendo la corretta gestione delle relazioni gerarchiche e dei vincoli richiesti.

La visualizzazione dell'organigramma e l'interazione dell'utente con esso sono gestite dal frontend, che aggiorna dinamicamente la rappresentazione in seguito alle operazioni effettuate. Le operazioni di eliminazione rispettano i vincoli definiti, impedendo stati inconsistenti (ad esempio eliminazione di unità con figli o assegnazioni).

I requisiti non funzionali sono soddisfatti grazie a un'interfaccia che guida l'utente durante le operazioni, prevenendo azioni non valide, e a una gestione persistente dei dati che garantisce la conservazione delle informazioni anche dopo il riavvio dell'applicazione. La separazione dei livelli e la modularità dell'architettura contribuiscono inoltre a migliorare la manutenibilità e l'affidabilità del sistema.

Il prototipo realizzato rappresenta una prima versione funzionante del sistema di gestione dell'organigramma aziendale.

L'obiettivo principale del prototipo non è quello di fornire un'interfaccia grafica definitiva, ma di validare i requisiti funzionali e non funzionali del progetto e dimostrare la correttezza dell'architettura proposta.

Il sistema permette di modellare una struttura organizzativa gerarchica composta da unità aziendali, visualizzandone le relazioni in modo semplice e immediato.

